oneM2M Architecture Based IoT Framework for Mobile Crowd Sensing in Smart Cities

Soumya Kanti Datta, Rui Pedro Ferreira da Costa, Christian Bonnet, Jérôme Härri
Mobile Communications Department, Eurecom, France,
Emails - {dattas, ferreira, bonnet, haerri}@eurecom.fr

Abstract—**The futuristic smart cities must have the capabilities to withstand the growing challenges on the urban infrastructure in terms of public safety, resource management, co-operative mobility management and more. To tackle these challenges, the cities are increasingly using next generation information and communication technologies (ICT). A plethora of the ICT based innovations are taking place on a wide range of domains – (i) cloud and mobile edge computing, (ii) sensing and actuation, (iii) low power communication, (iv) mobile crowd sensing and (v) big data analysis. These can be united under the umbrella of IoT and Machine-to-Machine (M2M) Communications. This paper introduces an efficient MCS enabled IoT framework for the smart cities. Co-operative crowd sensing coupled with a data centric approach can provide a uniform mechanism to mitigate many challenges faced by the smart cities. We also discuss integration of the mentioned IoT framework into oneM2M standard architecture. The three main contributions of the work are – (i) inclusion of a power aware mobile application development framework making crowd sensing applications context aware and (ii) a data driven approach for co-operative crowd sensing that creates actionable intelligence from raw data and (iii) deployment capabilities at cloud and edge computing platforms.**

Keywords—Mobile crowd sensing; Mobile application; IoT framework; oneM2M architecture; Smart city.

I. INTRODUCTION

Recent years have witnessed a significant research and industrial efforts towards the Internet of Things (IoT), Machine-to-Machine (M2M) Communications and Smart Cities. In an IoT ecosystem, the physical M2M devices are equipped with computational power, low power wireless technologies, software agents for data and configuration exchange. Thus things like sensors, actuators and RFID tags can connect to the Internet, communicate with peers, M2M gateways and cloud platforms to offer novel services to consumers. The IoT application scenarios extend to personal health monitoring, home control, smart grid, smart traffic & transportation management, smart environmental monitoring and more. The applications of IoT and M2M communications in the context of smart cities are of particular interest. This is due to the fact that a smart city aims to provide the mentioned services to its citizens. It is estimated that the urban population will grow by an estimated 2.3 Billion over next 40 years and around 60-70% of world population will reside in urban and surrounding areas [21]. This rapid growth will pose numerous challenges in terms of environmental changes, traffic congestion, access to healthcare, management of energy. A possible way to address these is through an effective mobile crowd sensing (MCS) mechanism [11]. MCS typically involves three main steps – (i) data generation and communication, (ii) data processing to derive high level abstraction and (iii) dissemination of knowledge to interested consumers which are similar to the basic steps of IoT. Therefore, it can help realize an effective MCS platform. To realize MCS enabled IoT framework, huge volume of sensors will be deployed to the upcoming smart cities. At the same time, the penetration of smartphones and tablets is substantially high in the consumer markets. Individuals equipped with such smart devices could personally or collectively contribute sensing information in a secured fashion to a public platform which can measure and infer additional information of interest or actionable intelligence. Traditionally such MCS systems are deployed for environmental monitoring, infrastructure monitoring, intelligent traffic management and social computing [16].

To utilize the availability of physical devices and high penetration of smartphones, we propose an IoT enabled MCS framework based on the oneM2M standard architecture [23]. We also present a power aware mobile application development framework for the application developers. This is due to the fact that continuous access to the smartphone sensors is expensive in terms of battery and CPU load. Majority of current mobile applications used for gathering data for crowd sensing drains battery very quickly making the consumers non-interested in MCS scenarios. Our framework allows development of self-adapting mobile applications which automatically adjust their behavior based on available battery and context information. Uniform sensor data exchange is performed using Sensor Markup Language which can be easily encoded into different media types. We employ semantic web technologies to reason on the sensor data and deduce new information of interest. This data driven approach makes the semantic computation independent of the underlying computing platform. This in turn allows the computing mechanisms to be deployed at a cloud platform or edge platform or even at M2M gateways. This flexibility promotes interoperability and robustness of the architecture.

The contributions of the paper are as follows – (i) an IoT framework for MCS which is integrated into an oneM2M standard architecture, (ii) the power aware framework for mobile application development for sensor data collection, (iii) deployment capabilities at cloud and edge computing platforms as well as M2M gateways, (iv) fusion of sensor data coming from multiple and heterogeneous domains paving way for co-operative crowd sensing applications and (v) examining the scalability of the overall architecture. Rest of the paper is organized as follows. Section II critically reviews the state-of-the-art and highlights the limitations in current MCS systems. Section III describes the power aware mobile application development framework. Section IV presents the MCS enabled IoT framework, its integration in the oneM2M architecture and its prototyping experience. Finally, the paper concludes by summarizing the contributions and future research directions.

II. STATE-OF-THE-ART

A plethora of research works have concentrated on mobile crowd sensing using participatory and opportunistic sensing methods. These methods, their applications and limitation are described below.

A. Crowd sensing paradigms

Authors Ganti, Ye and Lei [1] defines mobile crowd sensing as a paradigm "where individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest". The availability of more resources in mobile phones and their high penetrations into consumer markets have opened the vistas for novel MCS applications. They have classified the applications into environmental, infrastructure and social applications. Instead of sending raw sensor data to the cloud systems, localized analytics could be employed to infer the sensor data. This reduces the load on cloud and saves bandwidth. The authors also provided a deeper analysis on preserving the privacy of participating users by employing adequate security and data integrity mechanisms. At the end, an architecture and necessity of aggregate analytics for large scale study is also pointed out. Gaonkar et al [2] have presented a micro-blog, a platform for sharing and querying multimedia content through smartphone user participation. These multimedia contents are further enriched using smartphone sensors (e.g. accelerometer). Each blog is then associated with spatial and temporal information before being transmitted to an external web server. The position information can be used to map these blogs on maps like Google Map. Third party web services can be applied on these blogs and this creates the platform for participatory sensing. The web server allows end-users to issue queries to smartphones located at a specific position. Depending on the nature of the query, the responding smartphone can automatically sense and provide the response or it might require human involvement. The authors also illustrate several applications (e.g. news, alerts, social collaboration) utilizing their platform. They have also put forward a software architecture that considers energy aware localization techniques and applications, incentives for responding users and location privacy. A detailed review on participatory sensing can be found in [3]. It mentions the use of both of the network attested context and the physical context of the sensor data. This paper also advocates for the discovery of resources, globally reachable naming schemes and aggregation & dissemination policies for the data sources.

A remote sensing platform using smartphones is described by Das et al [4]. They propose to exploit the built-in capabilities of smartphones to carry out participatory sensing. A key challenge is to rapidly develop a mobile application. Instead of developing an application each time and make the effort to advertise it for wide adoption, the authors present their platform PRISM. It tackles generality, security and scalability issues associated with participatory sensing from individual smartphones. Generality takes care of the fact that the application developers can package their executable codes inside a generic application offering great flexibility. An innovative aspect of PRISM is resource metering which tracks the resource depletion in smartphones. Network, sensor, storage and energy usage are monitored and tightly controlled in order to avoid rapid depletion of resources in voluntary smartphone user participators. Another instance of participatory sensing is found in [5]. This paper considers the problem of road condition monitoring and traffic congestion in a city. The authors present a system called Nericell [5], capable of sensing audio, GPS and accelerometer sensors of smartphones carried by end-users. The collected sensor traces are further processed to calculate the accelerometer direction, breaking detection, stopping for pedestrian, bump detection. Although accessing many sensors (GPS and accelerometer) are continuous, the paper does not present any battery aware application development. Authors Cornelius et al have put forwards a privacy aware architecture Anonysense [15] for anonymous collaborative and opportunistic crowd sensing. The architecture allows end users to query and receive related response using a specific task language. It basically exploits the sensor data from the smartphones but paying great deal of attention to privacy. The authors discuss about a threat model assuming a rouge entity wants to de-anonymize participating carriers using eavesdropping and other mechanisms and alter data integrity. Consequently, the trust model is constituted on mobile nodes (i.e. end users), access points, registration authorities, and the applications running in mobile phones. The paper also reports on tasking and reporting protocols used in query and response.

B. Programming framework for crowd sensing

In order to harness the power of crowd sensing, a programming framework called Medusa is discussed in [13]. The motivations and requirements behind such framework are also highlighted. The system architecture introduces Medusa as a high level programming language in which the steps of crowd sensing are specified in terms of stages. The runtime is based on three principle designs – (i) partitioning of services between the smartphones and cloud systems, (ii) optimizing tasks running on smartphones, (iii) opt-in/opt-out policy for data transfer from phones to the cloud. Authors Ravindranath, Thiagarajan, Balakrishnan and Madden developed a system called "Code in The Air" (CITA) [14] to lower the barrier between programming and executing tasks. CITA makes it easier for developers to express conditions to reference high level activities of users. This is done in the activity layer using place hierarchies and activity compositions. The tasking framework of CITA enables the developers to write the tasking scripts using web technologies like Java Scripts. These tasks are then added to the user interface of the smartphones. The end users can then combine these tasks to create complex tasks.

C. *Applications of MCS*

More of such platforms exist in current literature which (i) assess personal environment through participatory sensing [6], (ii) provide searching facilities for multimedia contents [7], (iii) assists in locating parking spot for vehicles [8], (iv) collaborate with Twitter for text classification and trend analysis using crowd-sourced sensing [9], (v) answer certain queries with crowd-sourcing [10] and (vi) count the number of speakers at the certain location [12]. But these applications have been built using different architectures and programming frameworks. Interoperability among these architectures has not been investigated in the literature.

D. *Limitations*

A closer look into the state-of-the-art highlights the fact that mobile crowd sensing paradigm is still in its infancy. Although MCS has enormous potential and several use-cases in smart cities, following are limiting the scopes of it.

- There is no uniform method of collecting and sharing sensor data from smartphones and physical things. There is also a lack of standardized way of describing the sensor data coming from heterogeneous sources.
- Several MCS applications depend on continuous stream of sensor data from consumer smartphones. Access to sensors in smartphones and tablets is costly in terms of battery and processing power. There is no power aware mobile application development framework available for the developers.
- The mechanisms to interpret the sensor data are not uniform and not interoperable. It prevents MCS platforms in interacting among each other.
- The crowd sensing is limited to the sensors built-in to the smartphones. They do not interact with any sensors external to the smartphones.
- Current MCS applications are largely domain specific. Cross domain MCS applications are explored in a limited fashion in current literature.
- As identified in [3], such mobile participatory crowd sensing explicit, public and globally reachable naming schemes, discovery techniques which are yet to be developed and deployed to the smart cities.

III. POWER AWARE MOBILE APPLICATION DEVELOPMENT

Current mobile applications that communicate sensor data to MCS platforms are not power aware [19]. They continue to drain the battery even when it is critically low. In other words, these applications do not react to the remaining battery level and status. To mitigate this problem, we have designed and developed a framework to create power aware mobile applications which dynamically adapt their behavior based on available battery level, charging and context information at the real time. This is a novel aspect of the paper in the context of MCS. The framework is depicted in Figure 1 [17].

The framework is composed of – (i) battery and context monitoring engine, (ii) analyzer engine and (iii) self-adaptive features. The monitoring engine is collecting the remaining battery level, battery status (charging or discharging) and contextual information at real time from the smartphone. These are stored in a statistics module which computes a charging pattern. The context information contains spatial and temporal information. This determines if the user is travelling abroad (roaming network). The analyzer engine receives the battery and context information which are evaluated with respect to several pre-configured rules to determine an appropriate self-adaption profile. Each profile adapts the behavior of the application in terms of hardware & software resources, user features and any additional requirements.

The light self-adaptive profile is triggered when the battery level is sufficiently high (60-100%) and the user is not roaming. During the course of this profile, the mobile application can retrieve the smartphone sensor measurements as frequently as requested. The location information could be retrieved by GPS to be accurate. The developers could provide adequate user feature. The requested sensor data is converted to Sensor Markup Language (SenML) format (to settle the heterogeneity of sensor metadata) and communicated promptly over any wireless network.

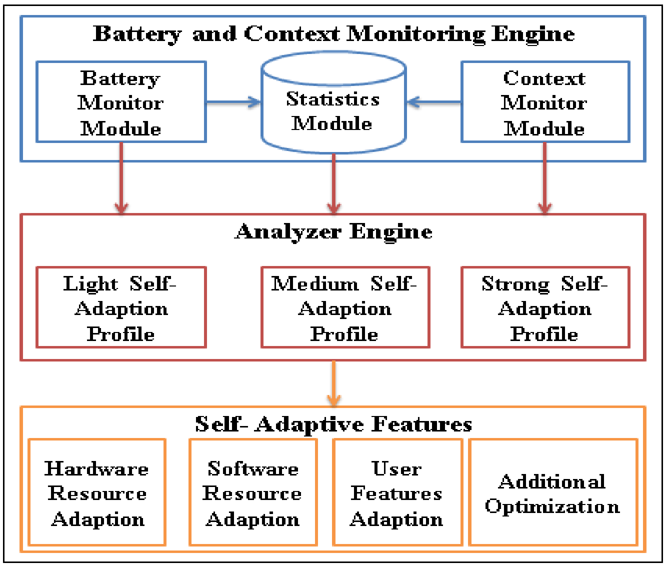

Fig. 1. Power aware mobile application development framework.

The medium self-adaptive profile starts when battery level is between 20% - 60%, battery status is discharging or USB charging and user is not roaming. This profile optimizes the access to hardware & software sensors and user experience level. The location information could still be obtained from GPS. Network access is preferred to Wi-Fi over mobile data. The user is given explicit option to choose which sensor data to be communicated to the MCS platform. Request about any other sensor data is not answered by the mobile application. The mobile application can also reduce the SenML metadata content if necessary. This leads to limited access to sensors which saves power in the smartphones. To reduce network operations, the mobile application can also bundle several SenML sensor data together and upload the bundle to the MCS platform.

The strong self-adaptive profile is activated when battery level is below 20% and discharging and/or if the user is roaming. In this case the user can configure the mobile application to not to contribute any sensor data to the MCS platform. The main aim of this profile is to save battery for the smartphone. This is a value addition for the end users as the mobile application is aware of the battery and context information. And the application can modify its behavior while accomplishing the MCS objectives.

IV. IOT FRAMEWORK FOR MOBILE CROWD SENSING

This section describes the proposed IoT Framework for MCS, its elements and its integration into oneM2M standard architecture. This is another novel aspect of the paper which ensures interoperability of the MCS platform with heterogeneous sensor devices, similar platforms and consumer centric services. Figure 2 depicts the framework. Following the high level requirements of crowd sensing scenarios, the framework includes – (i) a generation subsystem that contains physical things (vehicular sensors, environmental sensors) and smartphone which generate sensing information, (ii) a network subsystem to communicate the sensor metadata, (iii) a processing and storage subsystem generates high level knowledge from the raw data and (iv) a consumer subsystem which receives the knowledge taking advantage of the underlying MCS platform. A closer look at its building blocks for MCS is shown in Figure 3.

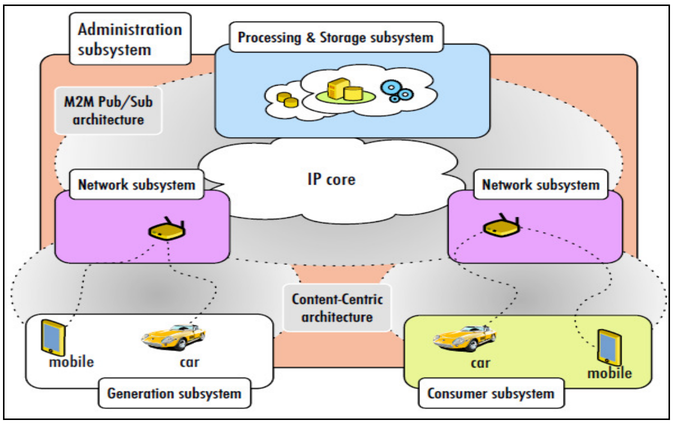

Fig. 2. IoT framework enabling mobile crowd sensing scenarios.

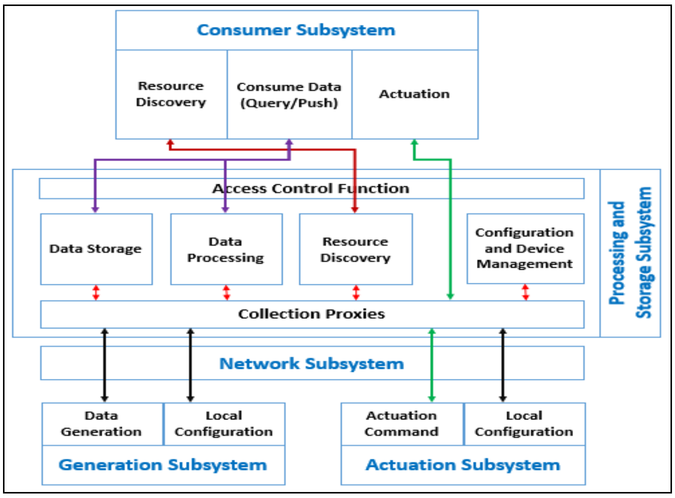

Fig. 3. Elements enabling mobile crowd sensing in IoT framework.

A. *Generation subsystem*

This is the very basis of MCS scenario since it provides the sensing information. Current literatures do not discuss any uniform structure for exchanging the sensor measurements in the context of MCS. Another point is that the sensor measurement alone does not allow intelligent computation and additional information (e.g. unit, type, timestamp, context) is required to infer high level knowledge. The generation subsystem exploits SenML to encode the measures as well as sensor attributes using JSON and settles the heterogeneity at sensing level. The local configuration element describes the device [22] and its sensors as well as communicating their descriptions to the processing and storage unit for the purposes of registration and discovery. Authorized consumers can access the local configuration and update it. The device and sensor descriptions are represented using CoRE Link Format [18].

B. *Network subsystem*

The network subsystem supports both IP and non IP based communication among other subsystems. Technologies such as ITS-G5 can be utilized as a medium for vehicle to infrastructure (V2I) communication. This allows MCS platforms to receive data from vehicular sensors. But there must be elements providing protocol translation since rest of the framework is IP based. Data generated in smartphone sensors are exchanged over TCP/IP networks.

C. *Processing and storage subsystem*

MCS platforms must derive meaningful high level abstraction from raw sensor data. This transformation takes places in this element of the IoT framework. Sensor metadata originating from heterogeneous sources and domains are treated using semantic web technologies. This also allows fusion of sensor data from different domains in the MCS platforms leading to co-operative crowd sensing applications. An example for the steps of sensor data fusion using semantic web technologies is shown in Figure 4.

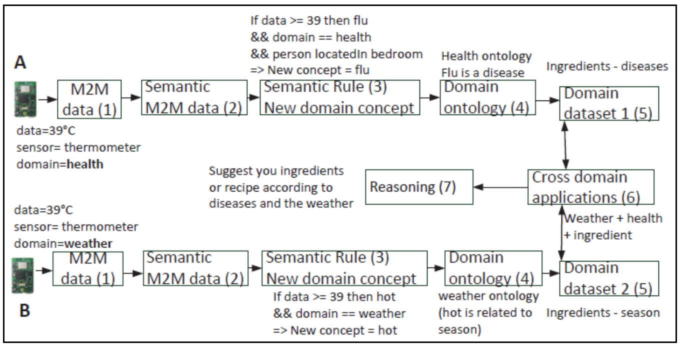

Fig. 4. M3 framework for fusion of raw sensor data using semantic web technologies [20], [24].

This approach exploits a Machine-to-Machine Measurement (M3) Framework for enriching the raw sensor data [20], [24]. It involves seven steps of processing. In the first step, the raw sensor data is encoded as uniform metadata following SenML recommendations. But it cannot be directly treated by semantic web technologies and must be converted in RDF format which is accomplished in second step. The third step applies semantic rules corresponding to the domain to derive a new concept. In the next step, domain ontology is applied to classify the derived concept. The fifth step involves utilization of domain datasets and paves way for creating cross domain application in the sixth step. Finally, semantic reasoning is used to fusion the knowledge derived in previous steps to create a completely new high level intelligence. Such fusion or combination of IoT sensor data from heterogeneous domains also highlight the data centric approach of the framework where the computation centers around the data and not on the infrastructure. This also makes the MCS scenarios and associated semantic computing independent of the underlying communication modules. As a result, the overall implementation becomes generic enough that can be deployed at a cloud or edge platform or even in an M2M gateway. In the MCS platforms, the processing and storage subsystem also includes additional elements for data storage, resource discovery [26], registration, configuration, device management [22], access control and security.

D. *Consumer subsystem*

The consumer subsystem receives the previously generated high level intelligence. It allows the consumers or other IoT devices perceive their environment, taking intelligent decision and reacting through actuators. Apart from that, the authorized consumers can also discover physical things, update their configuration and select sensors whose data to be utilized in the crowd sensing platform. We have integrated the IoT framework for MCS into the oneM2M standard architecture as depicted in Figure 5. This is another significant contribution of the work as it allows the IoT framework to maintain interoperability with similar platforms. The functionalities of the framework are developed and exposed as a collection of web services to make use of the web standards and best practices [18].

E. *Integration in oneM2M architecture*

The physical things in the architecture constitute the generation subsystem. The IoT framework functionalities and their computational capabilities are independent of the deployment platform (cloud, edge platform or M2M gateway).

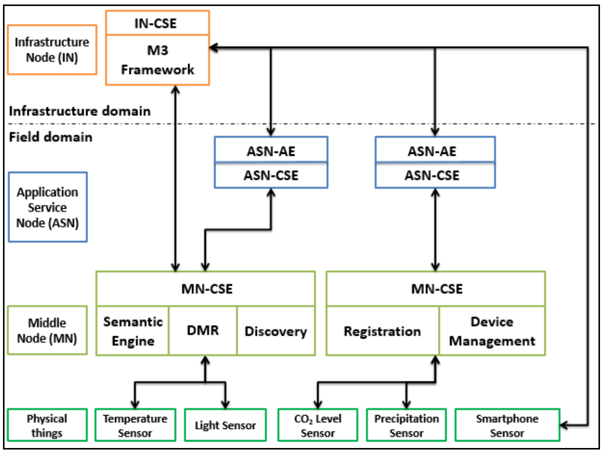

Fig. 5. oneM2M architecture integrating the IoT framework for MCS.

The common building blocks of the IoT framework include discovery, device management, registration, data management and repository (DMR), security and access control. In oneM2M architecture, these known as common service functions (CSF) and are a part of the common service entity (CSE). The middle node (M2M gateway or edge platform) as well as the infrastructure node (cloud platform) includes the CSEs and CSFs. But in the current oneM2M standards release [23], there is no semantic engine for enriching the raw sensor data. Therefore, to use the M3 framework in oneM2M, we have integrated a semantic engine as a CSF [25]. The architecture depicted in Fig 5 can deploy the MCS functionalities at both the middle and infrastructure nodes. Finally, the application service node (ASN) (i.e. the consumer subsystem) obtains the high level knowledge derived through the co-operative crowd sensing and data driven mechanisms. The power and context aware mobile application development logic is utilized in the generation subsystem when getting data from smartphone sensors as well as creating consumer mobile applications running in the ASN.

F. *Prototype development*

A brief summary of prototype of the oneM2M architecture is presented here (Figure 6). The MCS scenario chosen allows the resource discovery through any mechanism outlined in [26]. The provisioning phase allows the consumers or an administrator of the MCS platform to select the required sensors and their domains. Then the platform fetches the raw sensor metadata which passes through the convert, reason and query phase. This transforms the simple and raw sensor metadata into high level abstraction using the IoT application template housed in the M3 cloud. The high level abstraction on further treatment yields actionable intelligence that can provide some instructions to actuators to react to the environment.

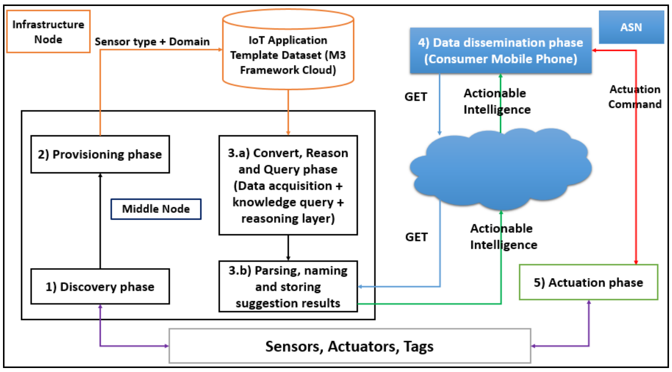

Fig. 6. Prototype development of the IoT architecture for MCS.

To maintain scalability and apply the overall architecture for dense MCS scenarios, the semantic computing part can be deployed into powerful edge servers or cloud systems. As a part of the performance evaluation, we have measured the memory footprints of the main software components. The device and sensor descriptions take less than 1KB. The memory footprint of discovery and provisioning mechanisms amount to 1MB as they are developed using a lightweight python scripting framework. The components and development framework of semantic computing requires a lot of memory and processing resources and are deployed in a Google Cloud Platform. The Android application for consumer devices consume around 10MB of internal memory. The lightweight implementation of the main software elements form stepping stone for enabling scalability and supporting dense MCS.

V. CONCLUSION

In a nutshell, this paper describes an IoT framework to enable MCS scenarios to solve the smart city challenges. Our extensive literature survey revealed the pros and cons of existing solutions. The first major contribution of the work is to provide a complete power and context aware mobile application development framework. Then the IoT framework, its subsystems and their integration into oneM2M architecture are discussed. The data driven approach stems from using the M3 framework which allows using semantic computing. Finally, the framework is generic enough to be deployed at any computing platform including cloud, edge or even M2M gateways. As of future work, we are examining a more focused implementation of the framework for smart city edge computing platforms and its performance evaluation.

ACKNOWLEDGMENT

This work is supported by French research project DataTweet (ANR-13-INFR-0008).

REFERENCES

[1] Ganti, R.K.; Fan Ye; Hui Lei, "Mobile crowdsensing: current state and future challenges," Communications Magazine, IEEE, vol.49, no.11, pp.32,39, November 2011.

[2] Shravan Gaonkar, Jack Li, Romit Roy Choudhury, Landon Cox, and Al Schmidt. 2008. Micro-Blog: sharing and querying content through mobile phones and social participation. In *Proceedings of the 6th international conference on Mobile systems, applications, and services* (MobiSys '08). ACM, New York, NY, USA, pp. 174-186.

[3] J. Burke and D. Estrin and M. Hansen and A. Parker and N. Ramanathan and S. Reddy and M. B. Srivastava. 2006. Participatory sensing. In Workshop on World-Sensor-Web (WSW'06): Mobile Device Centric Sensor Networks and Applications. ACM, pp. 117-134.

[4] Tathagata Das, Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee, and Asankhaya Sharma. 2010. PRISM: platform for remote sensing using smartphones. In Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10). ACM, New York, NY, USA, pp. 63-76.

[5] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. 2008. Nericell: using mobile smartphones for rich monitoring of road and traffic conditions. In Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys '08). ACM, New York, NY, USA, pp. 357-358.

[6] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Péter Boda. 2009. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys '09). ACM, New York, NY, USA, pp. 55-68.

[7] Tingxin Yan, Vikas Kumar, and Deepak Ganesan. 2010. CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. In Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10). ACM, New York, NY, USA, pp. 77-90.

[8] Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. 2010. ParkNet: drive-by sensing of road-side parking statistics. In Proceedings of the 8th international conference on Mobile systems, applications, and services (MobiSys '10). ACM, New York, NY, USA, pp. 123-136.

[9] Demirbas, M.; Bayir, M.A.; Akcora, C.G.; Yilmaz, Y.S.; Ferhatosmanoglu, H., "Crowd-sourced sensing and collaboration using twitter," World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a , vol., no., pp.1,9, 14-17 June 2010.

[10] Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. 2011. CrowdDB: answering queries with crowdsourcing. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (SIGMOD '11). ACM, New York, NY, USA, pp. 61-72.

[11] Cardone, G.; Cirri, A.; Corradi, A.; Foschini, L., "The participact mobile crowd sensing living lab: The testbed for smart cities," Communications Magazine, IEEE , vol.52, no.10, pp. 78-85, October 2014.

[12] Chenren Xu; Sugang Li; Yanyong Zhang; Miluzzo, E.; Yi-fam Chen, "Crowdsensing the speaker count in the wild: implications and applications," Communications Magazine, IEEE , vol.52, no.10, pp. 92-99, October 2014.

[13] Moo-Ryong Ra, Bin Liu, Tom F. La Porta, and Ramesh Govindan. 2012. Medusa: a programming framework for crowd-sensing applications. In Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys '12). ACM, New York, NY, USA, 337-350.

[14] Lenin Ravindranath, Arvind Thiagarajan, Hari Balakrishnan, and Samuel Madden. 2012. Code in the air: simplifying sensing and coordination tasks on smartphones. In Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications (HotMobile '12). ACM, New York, NY, USA, Article 4, 6 pages.

[15] Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. 2008. Anonysense: privacy-aware people-centric sensing. In Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys '08). ACM, New York, NY, USA, pp. 211-224.

[16] Yu Xiao, Pieter Simoens, Padmanabhan Pillai, Kiryong Ha, and Mahadev Satyanarayanan. 2013. Lowering the barriers to large-scale mobile crowdsensing. In Proceedings of the 14th Workshop on Mobile Computing Systems and Applications (HotMobile '13). ACM, New York, NY, USA, Article 9, 6 pages.

[17] Datta, S.K.; Bonnet, C.; Nikaein, N., "Self-adaptive battery and context aware mobile application development," Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International, pp. 761-766, 4-8 Aug. 2014.

[18] Datta, S.K.; Bonnet, C.; Nikaein, N., "An IoT gateway centric architecture to provide novel M2M services," Internet of Things (WF-IoT), 2014 IEEE World Forum on, pp. 514-519, 6-8 March 2014.

[19] Datta, S.K.; Bonnet, C.; Nikaein, N., "Android power management: Current and future trends," Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012 First IEEE Workshop on, pp. 48-53, 18 June 2012.

[20] A. Gyrard, S. K. Datta, C. Bonnet and K. Boudaoud, "Standardizing generic cross-domain applications in Internet of Things," Globecom Workshops (GC Wkshps), 2014, Austin, TX, 2014, pp. 589-594.

[21] United Nations, ''World Urbanization Prospects'', 2009.

[22] Datta, S.K.; Bonnet, C., "A lightweight framework for efficient M2M device management in oneM2M architecture," in Recent Advances in Internet of Things (RIoT), 2015 International Conference on, pp.1-6, 7-9 April 2015.

[23] TS-0001-oneM2M-Funtional-Architecture-V-2014-08. http://www.onem2m.org/images/files/deliverables/TS-0001-oneM2M-Functional-Architecture-V-2014-08.pdf

[24] A. Gyrard, C. Bonnet and K. Boudaoud, "Enrich machine-to-machine data with semantic web technologies for cross-domain applications," Internet of Things (WF-IoT), 2014 IEEE World Forum on, Seoul, 2014, pp. 559-564.

[25] A. Gyrard, S. K. Datta, C. Bonnet and K. Boudaoud, "Integrating machine-to-machine measurement framework into oneM2M architecture," Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific, Busan, 2015, pp. 364-367.

[26] S. K. Datta, R. P. F. Da Costa and C. Bonnet, "Resource discovery in Internet of Things: Current trends and future standardization aspects," Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, Milan, 2015, pp. 542-547.